

Representative Views and Paths for Volume Models

Pere-Pau Vázquez¹, Eva Monclús², and Isabel Navazo¹

¹ Modeling, Visualization, and Graphics Interaction Group
Dep. LSI, Universitat Politècnica de Catalunya (UPC) / Spain
{pere.pau, isabel}@lsi.upc.edu

² Institut de Robòtica i Informàtica Industrial, UPC-CSIC
emonclus@iri.upc.edu

Abstract. Volume data models are becoming larger and larger as the capture technology improves. Thus, their visualization requires high computational power. The automatic presentation of volume models through representative images and/or exploration paths becomes more and more useful. Representative views are also useful for document illustration, fast data quality evaluation, or model libraries documentation. Exploration paths are also useful for video demonstrations and previsualization of captured data. In this paper we present a fast, adaptive method for the selection of representative views and the automatic generation of exploration paths for volume models. Our algorithm is based on multi-scale entropy and algorithmic complexity. These views and paths reveal informative parts of a model given a certain transfer function. We show that our method is simple and easy to incorporate in medical visualization tools.

1 Introduction

Optimal selection of viewpoints is an important task in 3D visualization. The amount of information perceived by an observer is determined by the selected view and shading function. Although the visualization techniques are becoming faster and faster, the size of datasets also increases, and the time users are able to devote to a single analysis is limited. In this sense, we propose some techniques to help users to find adequate views of the datasets in an efficient way. The presented techniques are useful for general volume models although we mainly use medical examples.

Our proposal works upon a model classified through the definition of a transfer function (TF) and the specification of a region of interest. Starting from this minimal information, we generate both a set of representative views of the model and an exploration path that allows users to choose the most suggestive view and, optionally, modify it. The user may also modify the TF if needed, in order to refine the inspection, or to render the same exploration path showing different information. This method is useful, as it allows to obtain representative views in a short time, comparable to the loading time of the dataset, and permits the generation of inspection paths at almost no extra cost.

Automatic selection of informative views, as well as qualitative exploration paths are also useful for book or articles illustration, fast model previsualization, good viewpoints for scene introduction and, in general, as a tool to optimize access to the in-

teresting information and facilitate the understanding of the anatomic structures [1, 2]. Moreover, this can also be used for illustrating models libraries.

In this paper we present an approach for the automatic illustration of volume models via the selection of representative, perceivably different views of volume data, and for the generation of exploration paths around these distinctive parts of the models. The exploration path is built based upon the visually revealed information of the direct volume rendering process, which makes the navigation more closely related to the information that will be perceived by the user. We do not concentrate on TF design, which is orthogonal to our problem. Despite that, since our method emphasizes the information revealed by the TF, we may improve the knowledge of the model by performing the analysis (i. e. selection of representative views, and exploration path) under different TFs. The main contributions of our paper are threefold:

- The selection of a good view criterion that evaluates visual information for volume models. Based on it, we build a fast best view selection algorithm that uses an adaptive scheme. This best view can be chosen as a starting point for the exploration path for on-line inspection, thus reducing the human effort.
- A method based on Kolmogorov complexity that determines the set of visually different views of a volumetric model. These may form the representative set of views of a model, usable to illustrate model libraries or serve as key points for automatic exploration path construction.
- From the information gathered in the previous step, we build an exploration path around the model that reveals a high amount of information from it and passes through the visually different views of the model. This aids the user discovering the most important features of the model in a soft manner.

The main advantage of our method is that it does not require any preprocess. Moreover, it gives the results in a time comparable to loading a model, and thus, the user can immediately begin the inspection starting from a good view of the model.

Next section introduces previous work on view selection for surface-based and volume-based datasets. Section 3 presents the visual quality evaluation measure we selected and our adaptive best view selection algorithm. In Sect. 4 we develop a method for the selection of representative views based on Kolmogorov complexity. Then, Sect. 5 gives details on how to build an exploration path around an object that reveals a high quantity of information from it. We discuss our results and summarize our work in Sect. 6.

2 Previous Work

Viewpoint selection has been addressed with different focuses in different areas. In surface-based approaches, the good viewpoints are selected taking into account the geometric information of the models. Usually, three parameters are taken into consideration: i) the number of visible faces, and ii) the area of the object or the visible faces' projections, and iii) the projected silhouette of the object. The analysis is focused under heuristic functions [3] or information theory-based approaches [4]. Several of these methods have been analyzed by Polonsky *et al.* [5] who conclude that none of them can

be coined as *universal*. On the contrary, one can always find some object whose best view is *missed* by the developed metrics.

Previous approaches for best view selection of triangular models are not amenable to direct volume rendering because the rendering result is not a set of polygons, and because the usual rendering techniques produce images where, for each pixel, more than a single (iso)value contributes to its final color. Thus, some methods have been developed in order to analyze volumetric information, such as the ones by Bordoloi and Shen [6], Ji and Shen [7], and Takahashi *et al.* [8]. These methods take into account the information contained in the dataset, and thus require a previous preprocess, in some cases quite costly. Furthermore, they do not measure quality based on the rendered views, and thus, different structures might be treated separately although they could produce a uniform colored region on screen. In contrast to these, our approach works on the generated image, with the objective of measuring only the information that will be effectively seen by the user. Mühler *et al.* [1] focus on intervention planning. They preprocess a set of viewpoints placed at 4096 positions on a bounding sphere. At each point, a set of parameter maps is computed that indicate the influence of the current quality parameter settings on the viewpoint. From this information, by using weighted parameter maps that are application dependent, the authors get the information needed to generate cutaway views of the objects of interest. Viola *et al.* [2] have also developed a preprocess method which takes into account, not only a viewpoint quality metric, but also information on focus of attention. Our objective is somewhat complementary to those, as we focus on the fast selection of representative views and exploration paths of models, without the need of segmenting the dataset and minimizing the manual interaction, which is possible thanks to the viewpoint quality measure we selected. However, we may also analyze selected regions of the model as shown in Fig. 7-left.

A closely related problem is the analysis of the information contents in an image. It does not take into account the geometry of the scene but the features of the rendered view. Image analysis techniques have been developed for the automatic selection of lighting configuration from an inverse engineering fashion [9], or with direct image measurement processes [10–12]. Those approaches take into account the final image, and measure the information using perceptual or information-theoretic measures. For the view quality evaluation of volume models, we use the approach by [12], presented next, that analyzes the rendered images with a multiresolution entropy measure, due to its simplicity, and the lack of manual configuration required.

3 Fast Best View Selection

As already commented in the previous section we aim at analyzing the information that finally arrives at the user given a certain transfer function. From the metrics we analyzed, the one based in Multi-Scale entropy seems to be adequate for our purposes, as is simple, easy to evaluate, and requires no user intervention. We will show that this metric, when applied to volumetric data renditions, leads to good views and it is robust to changes in the resolution of the rendered view. Next, we justify our viewpoint quality metric and we explain how to use it for interactive good viewpoint selection.

3.1 Multi Scale Entropy

Typically, the amount of information in an image has been measured using Shannon entropy or one of its derivatives. Shannon entropy gives the average *information* or the *uncertainty* of a random variable:

$$H(X) = - \sum_{i=1}^N p_i \log p_i, \quad (1)$$

where $X = \{X_1, X_2, \dots, X_N\}$ is an image containing integer values and N is the number of different values of a pixel and p_i are the values obtained from the histogram of X , that is, the probabilities of each histogram entry. The logarithms are taken in base 2 and $0 \log 0 = 0$ for continuity; $-\log p_i$ represents the *information* associated with the result X_i . The unit of information is called a *bit*. Observe that this entropy will be 0 if all the pixels have the same value and maximum when all pixels have a different value.

Most of the entropy-based measures for images rely on histogram analysis which, unfortunately, are insensitive to the correlation of pixels. To overcome these problems, Starck *et al.* introduce the concept of multiresolution into the entropy [13] (a.k.a. *multi scale entropy*). They consider that the information in an image is the sum of the information at different resolution levels l . This can be achieved by using a wavelet transform W of an image. Thus, the image information can be measured using the wavelet coefficients $w_{l,k}$ for a fixed set of levels. If the number of levels is high enough, the remaining information can be considered background. The amount of information produced by a rendering can be measured by analyzing the Shannon entropy of wavelet coefficients of each color channel at each level:

$$H_W(X) = - \sum_{l=1}^L \sum_{k=0}^{N_l} h_{RGB}(w_{l,k}), \quad (2)$$

where $h_{RGB}(w_{l,k})$ is $-p(w_{l,k}) \log p(w_{l,k})$, with $p(w_{l,k})$ being the relative number of coefficients of the channel with value k in level l . Hence, h_{RGB} means that the entropy is measured separately for each RGB channel. We have applied this method and found that it is suitable for volume rendering analysis because it reasonably analyzes the structures given a TF, as shown next. In [12] a user study showed that the Multi Scale entropy was a good approximation to the amount of information revealed by illumination.

3.2 Interactive Good Viewpoint Selection

The use of the described measure for good viewpoint selection of volume models gives good results, as shown in Fig. 1 and Fig. 2. Here, the quality has been measured for a dense set of viewpoints around a model, and we can observe how it computes higher values (warmer and larger spheres, being the pink sphere the best viewpoint) where more details are provided from the object. Unfortunately, for the purpose of fast model previsualization, a brute-force approach is not practical, because the analysis of a sufficient set of views takes roughly half a minute. As the loading times of the models take several seconds (see Table 1) on a fast machine (a QuadCore PC with 8GB of memory

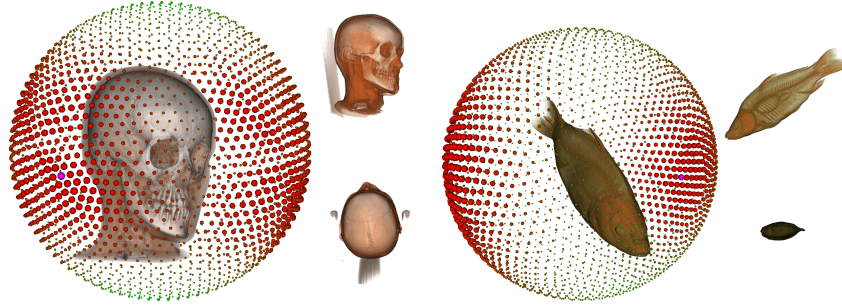


Fig. 1. The quality values for a dense set of views around the head and the fish models. Viewpoint quality is encoded in both the color (the higher the quality the warmer the color) and the node size (the higher the quality the larger the size). The best view node is painted pink. The images placed on the right of each view quality sphere show the best (top) and worst (bottom) views according to our measure.

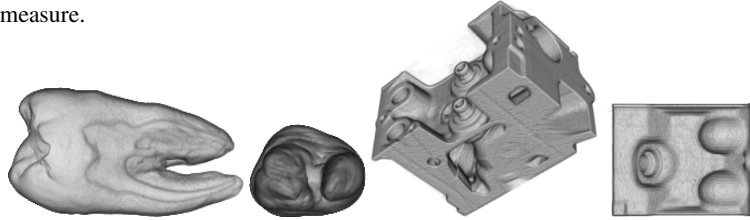


Fig. 2. Best and worst views for the tooth and the engine models.

with a GF 8800 card), it seems acceptable to have a good viewpoint selection algorithm that takes at most the loading time. Note (cf. Table 1) that all timings depend on the dataset dimension, number of analyzed views, and the transfer function applied (which determines the number of processed voxels), that is why a smaller model may have a worse timing than a larger one.

In order to achieve better efficiency than an exhaustive search, we propose two improvements:

Image resolution: We have analyzed and compared the quality measure for a dense set of viewpoints (2562) on a bounding sphere in order to evaluate how the results depend on the resolution of the viewport. We analyze the images generated offscreen at smaller viewports than the used for regular rendering. Our analysis showed that resolutions of 256×256 are good enough for best view computation for all the models we tested, as the best views were identical to the ones selected with 512×512 images. We use 512^2 viewport for high quality rendering, as this size is comparable to the resolution of the models we have. Smaller images, such as 128×128 , generally produce good views (very similar if not equal to the optimal) for most models and thus can be used if a quickly response is mandatory (see Fig. 3).

Adaptive search: In order to achieve interactive rates, we also perform the best view search in an adaptive fashion, starting from a coarse sampling of the bounding sphere and adaptively subdividing it according to an estimator of the entropy.

Our algorithm builds a spherical triangle mesh with the vertices placed at the vertices of an icosahedron. For each vertex i , the quality measure H_i and the total maximum H_{max} is evaluated and stored. Similarly to Gumhold [11], we treat the quality function as Lipschitz-continuous in a neighborhood of the analyzed points and seek for the Lipschitz constant L of the mesh. This value gives an estimation of the maximum quality variation around a viewpoint.

Our algorithm consists of two steps: First, for each triangle of the current mesh, the maximum reachable entropy point at each edge ($He_1 \dots He_3$) is estimated using L (see Fig. 4a). Then, we estimate the maximum value inside the triangle He_{max} by interpolating the computed values (He_1 to He_3). If $He_{max} + K_{safe}$ is higher than H_{max} , the actual value is measured by rendering a new viewpoint from this position and adding the subdivided triangle (see Fig. 4a). We add a safety constant K_{safe} in order to be conservative and avoid missing any maximum. We found experimentally that this constant is only necessary when the range of entropy values of the initial subdivision is not very large, which makes the Lipschitz constant small (below 1). Therefore, we add a $K_{safe} = 0.02 * H_{max}$ when $L < 1$. Each time a new viewpoint is added, the Lipschitz constant is recalculated. Our algorithm stops when none of the estimated entropy values is higher than H_{max} , or when those views are too close (i. e. 5 degrees) to existing analyzed positions. Figure 4b shows an example of the subdivision produced for the feet model.

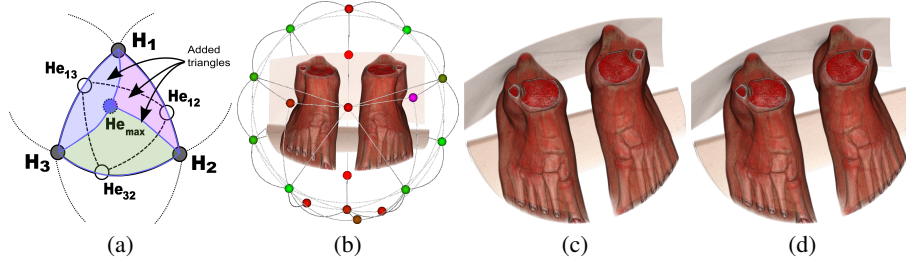


Fig. 4. Left shows the adaptive subdivision algorithm. On the right we can see the result of two steps of the subdivision for the feet model. The best view of the feet model as selected among 2562 views (c) and our adaptive method (d).

With this approach we obtain very similar maximum values than the ones obtained with the brute-force method, but at a fraction of the time (see Table 1). As an example, we can see in Fig. 4-right two views of the same model, the left one was selected as the best of 2562 views, and the second one with our adaptive method, although the

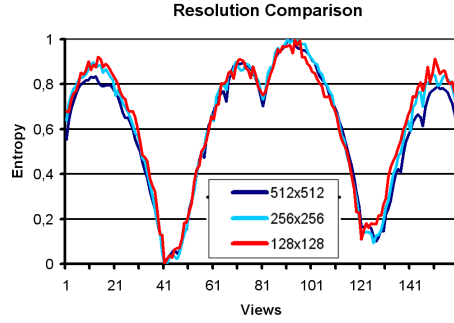


Fig. 3. Viewpoint quality evaluation of the head model under different resolutions: 512×512 , 256×256 , and 128×128 . The camera has covered a circular path around the X axis. Note the high similarity between all the resolutions.

positions are not exactly the same, the information shown is almost identical. Note that, for the best view computation, a lower number of regularly placed viewpoints is usually enough, being 162 a good compromise, and therefore this is the number of views we use in the timings shown in Table 1 for the brute force approach.

Model	Slices	load time (s)	Best view 512 ² / 256 ² / 128 ²	Adaptive 256 ² #views / time	Adaptive 128 ² #views / time	Represent. views 256 ² / 128 ²
Head	486	12.09	36.41 / 13.47 / 5.20	28 / 2.32	27 / 0.94	3.85 / 0.40
Torax	291	4.50	25.66 / 8.43 / 3.14	34 / 1.79	26 / 0.56	2.70 / 0.54
Feet	282	4.49	34.2 / 11.13 / 4.03	26 / 1.85	23 / 0.59	1.93 / 0.42
Coxis	100	1.69	18.89 / 5.48 / 2.09	34 / 0.9	18 / 0.26	2.56 / 0.17

Table 1. Comparison of the computation times for different models compared to the loading time. All models have resolution $512 \times 512 \times \text{slices}$, and *slices* are indicated in column 2. Fourth column shows the seconds required to compute the best view by using a set of 162 positions. The fifth column shows the time needed to obtain the best view with our adaptive method. The last column shows the time needed to find 3 representative views for the model. In contrast to previous columns, where the cost is lineal with size and number of views, here the cost also depends on the compression process which behaves differently for different files of the same size.

The time needed to compute the best view using our adaptive method is depicted in Table 1 for different models. Note how, in all cases, we may compute the best view in, roughly, 1/4th of the time needed to load a model, and with low resolution viewpoints (i.e. 128²) even faster.

4 Representative View Selection

When illustrating a complex model, a single view (the one with the highest entropy value) may be insufficient, for example because some details of the object may still be missing or because it does not provide enough information from the 3D relationship of the structures. Selecting views restricting to the ones with high entropy is generally not a good choice because this may lead to show similar information under different directions. This is specially true with models that have a certain level of symmetry. Our brain is very efficient in inferring symmetries, and therefore, it is more valuable to get a set of views that show information not captured in the best view. Although some of them might not be the most informative ones, they will serve as a complement and allow a better understanding of the relationships between the different structures of the model.

Previous approaches require the knowledge of structures present in the dataset, which are obtained via a segmentation process. We would like to avoid this pre-process. In order to do that, we propose an approach that consists in directly compare the images obtained in the process explained in Sect. 3.1. The view set that adequately represents a model is determined with a greedy scheme:

1. Select the best view B_0 with the adaptive algorithm.
2. Measure the distances from B_0 to the remaining views.

3. Next representer view B_1 is the one at the highest distance from B_0 .

The algorithm cost is linear with the number of views, being the comparing process the most costly one. In order to determine the distance between two views ($d(X, Y)$), we use the Normalized Compression Distance as explained in the following sections.

Once we have the two initial representative views, if we want to gather the missed information by these two, we can proceed the same way: We compute the distances from the remaining views against B_1 and choose as new view X the one that maximizes the geometric average of the distances to B_0 and B_1 . This process can be repeated several times, but three or four are usually enough for most models. In order to find the optimal set of M views, a global maximization could be applied, but would have a quadratic cost in the number of views and we obtain similar results with our approach.

At this point, the remaining work is to select a robust view similarity measure. Bordoloi and Shen [6] use the Jensen-Shannon divergence over the visible voxels, and do not take advantage of potential symmetry of the dataset, which is desirable. Nonetheless, we cannot compare views by using simple image metrics such as Mean Square Error, as those metrics are sensitive to transformations of the image such as rotation or translation, which may become a problem when comparing two views automatically generated for model inspection. In order to solve these problems, we propose a conceptually simple solution with foundations in algorithmic complexity, concretely, we evaluate view likelihood with the *Normalized Compression Distance*.

4.1 Normalized Compression Distance

Normalized Compression Distance is a universal metric of distance between sequences. It has its roots in Kolmogorov complexity (also known as algorithmic complexity). We briefly detail here some concepts of algorithmic complexity. The interested reader can refer to Li and Vitányi's book [14] for a deeper and more theoretical introduction.

The **Kolmogorov complexity** of a file is the ultimate compressed version of the file. Formally, *Kolmogorov complexity* ($K(x)$) of a string x is the length of the shortest binary program to compute x on a universal computer (such as a universal Turing Machine). Thus, $K(x)$ denotes the number of bits of information from which x can be computationally retrieved. Hence, $K(x)$ is the lower-bound of what a real-world compressor can possibly achieve.

The **conditional Kolmogorov complexity** $K(x|y)$ of x relative to y is the length of a shortest program to compute x if y is provided as an auxiliary input. Both Kolmogorov complexity and conditional Kolmogorov complexity are machine independent up to an additive constant.

Given these two definitions, Bennet *et al.* [15] define the **information distance** between two, not necessarily equal length binary strings as the length of the shortest program that can transform either string into the other one, both ways. The information distance is a metric. Li *et al.* [16] present a normalized version of information distance dubbed *similarity metric*, defined as:

$$d(x, y) = \frac{\max\{K(y|x), K(x|y)\}}{\max\{K(x), K(y)\}} \quad (3)$$

The authors also prove that this metric is universal (two files of whatever type similar with respect to a certain metric are also similar with respect to the similarity metric). Being Kolmogorov complexity not computable, it may be approximated with the use of a real-world compressor, leading to the **Normalized Compression Distance (NCD)**:

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \quad (4)$$

where function $C(F)$ is the size of the compression of a certain file F , and xy is the concatenation of files x and y . Although the similarity metric has values in $[0..1]$, NCD values are usually in the range of $[0..1.1]$, due to compressor imperfections. NCD has been used for applications such as language classification and handwriting recognition [17].

4.2 NCD-Based Representative View Selection

Cilibrasi and Vitányi [17] analyze the conditions that compressors must fulfill in order to be used for computing the Normalized compression distance. They state that most of them do, such as stream-based (*zlib*), block-based (*bzip*), and statistical (*PPMZ*) compressors. As studied by Cebrián *et al.* [18], the use of compressor for comparison purposes requires some issues to be taken into account, for instance, the size of the compressed files limits the efficiency of the comparison processes. In the case of *bzip2*, the *best* option works properly for files up to 900KB before being compressed. Larger sizes make the comparison processes less effective. Based on this, we use *bzip2* compressor on PPM images. Note that the images we analyze are of 256×256 or 128×128 , and therefore, the size of the concatenated file never exceeds 900KB. One might wonder if specialized compressors (i.e. *JPEG* or similar) would do a better job than general compressors, but, we tested lossless compression using *JPEG2000*, which generated larger files (roughly 20%) than *bzip2*.

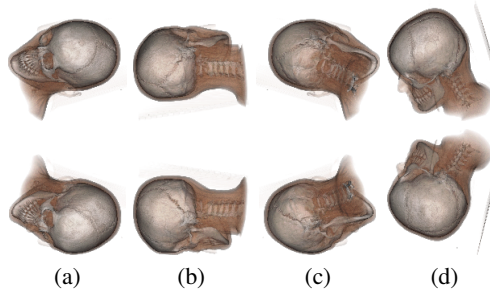


Fig. 5. From left to right, each column shows the pair of images whose distance is the smallest with respect to the rest. Note how our similarity measure is robust with respect to rotation and symmetry. Images are measured as is, that is, without any rotation to align them.

In consequence, to determine the distance between two images ($d(X,Y)$), we concatenate them, and then we compress the original and the concatenated files. Then, the distance is measured using (4). In Fig. 5 we show, from a set of views around the head model, the four pairs whose distance is the smallest. Note how symmetric views are

correctly ranked as being very similar. An example of selected representative views is shown in Fig. 6, where the three selected views for the head and fish models are shown. Observe that these views show substantially visually different information. Representative views computation is quite fast (see Table 1), being the concatenation and compressing process the most costly part. However, in most cases we still require a total time (adaptive best view selection + representative view selection) smaller than the loading time. Moreover, we may perform the whole process in roughly one second for most of the models tested if we restrict ourselves to offscreen-viewports of 128^2 .

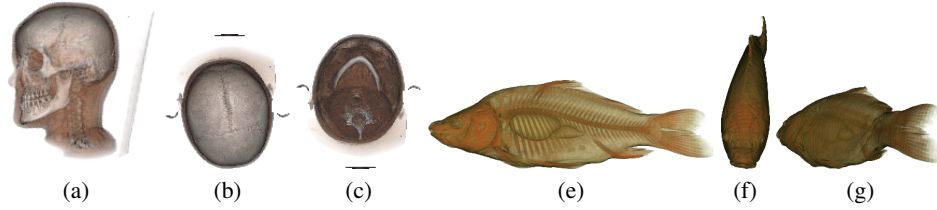


Fig. 6. The 3 representative views of the head and the fish model. Note that they properly represent different appearances of the models.

5 Exploration Path Construction

For complex objects, the construction of an inspection walkthrough may be very useful, as a set of views may be insufficient for providing a global vision of the model. Therefore, we propose an algorithm for generating exploration paths for model inspection. This path consists on a set of positions located on a bounding sphere of the object or region of interest. We use as key points the important viewpoints selected in the previous section and ensure visiting all of them. The viewpoint with the highest entropy is visited in second place, as this allows the camera to visit the surroundings of this point that, intuitively, should be more informative than the rest.

The path construction algorithm determines the minimal length path that passes through all the representative views and ensures we are providing a high amount of information. As we have only three to four candidates, an exhaustive search of the shortest path that ensures we pass through the best view in second position is computed instantaneously. We call this the *simple exploration* algorithm.

In order to maximize the information gathered through the exploration, we introduce an *improvement*: the camera may deviate from the short path up to a certain distance at each step. For each step, we analyze the entropy of three candidates, which are placed toward the destination point and separated by ± 4 to 6 degrees. The one with the highest entropy is chosen. The allowed total deviation is limited and reduced as long as we get closer to the next key point. This ensures we are visiting all keypoints. Furthermore, the speed of the camera is reduced when visiting points nearby the best one (B_0) because, intuitively, they will show a higher amount of valuable information. In Fig. 7 we can see the results with the *simple path* and the *improved path* for the exploration of the kidney and torax models. The improved approach gathers a higher amount of details than the

simple method. The *simple path* calculation time is negligible, but the *improved path* requires the evaluation of entropy at each step, and therefore, the time will depend on the number of steps we want to produce.

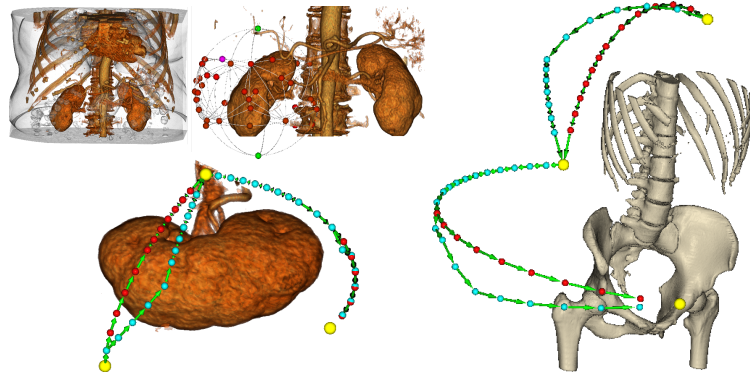


Fig. 7. At the left side: Analysis of a region of interest around the kidney. At right side: A comparison of the simple exploration path (red) and the improved version (cyan) for two models, the kidney and the thorax.

6 Conclusions

Throughout the paper we have analyzed and commented the efficiency and quality of the results obtained applying our techniques either for the good view selection and the exploration path construction. Its fast response, ease of use, and lack of parameter definition, facilitates its incorporation in daily work by i.e radiologists at almost null cost. Nonetheless, its simplicity makes it easy to incorporate in a medical volume visualization package but may also be applied to other rendering metaphors.

We may also apply our technique to a region of interest defined by the user, who also defines the proper TF, and then, our algorithm computes an exploration path around the region of interest. The result is shown in Fig. 7-left. Once the exploration path is obtained, we can modify the TF in order to emphasize different anatomic structures and help in understanding their spatial relationship (see Fig. 7).

In this paper we have explored different techniques for fast volume models exploration. We used an image information contents evaluation based on multi-scale entropy in order to determine good viewpoints for volume models using a fast adaptive method. We have developed algorithms for the selection of representative views of models and the generation of exploration paths. The key difference of our approach is that we use a metric on the generated views, which allows us to abstract from the transfer function or rendering method used. Similarity between views is evaluated by using the notion of *Normalized Compression Distance*, borrowed from Kolmogorov complexity field. Our method is useful, not only for volume model inspection, but also easy to incorporate to medical visualization tools, document illustration support, and automatic labeling of model libraries. One of the main advantages is that we do not require any preprocess, and therefore, our method is adequate for fast previsualization of models.

7 Acknowledgments

Thanks to "The Volume Library" at <http://www9.informatik.uni-erlangen.de/External/vollib/> for providing several datasets.

References

1. Mühler, K., Neugebauer, M., Tietjen, C., Preim, B.: Viewpoint selection for intervention planning. In: EG/ IEEE-VGTC Symposium on Visualization. (2007) 267–274
2. Viola, I., Feixas, M., Sbert, M., Gröller, M.E.: Importance-driven focus of attention. IEEE Transactions on Visualization and Computer Graphics **12**(5) (October 2006) 933–940
3. Plemenos, D., Benayada, M.: Intelligent display in scene modeling. new techniques to automatically compute good views. In: Proc. International Conference GRAPHICON'96
4. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: Proceedings of the Vision Modeling and Visualization Conference (VMV-01), Stuttgart (2001) 273–280
5. Polonsky, O., Patanè, G., Biasotti, S., Gotsman, C., Spagnuolo, M.: What's in an image? The Visual Computer **21**(8-10) (2005) 840–847
6. Bordoloi, U., Shen, H.W.: View selection for volume rendering. In: IEEE Visualization. (2005) 487–494
7. Ji, G., Shen, H.W.: Dynamic view selection for time-varying volumes. IEEE Transactions on Visualization and Computer Graphics **12**(5) (2006) 1109–1116
8. Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In: IEEE Visualization. (2005) 495–502
9. Patow, G., Pueyo, X.: A survey on inverse rendering problems. Computer Graphics Forum **22**(4) (2003) 663–687
10. Shackel, R., Lischinski, D.: Automatic lighting design using a perceptual quality metric. Computer Graphics Forum (Proceedings of Eurographics 2001) **20**(3) C–215–226
11. Gumhold, S.: Maximum entropy light source placement. In: Proc. of the Visualization 2002 Conference, IEEE Computer Society Press (October 2002) 275–282
12. Vázquez, P.: Automatic light source placement for maximum illumination information recovery. Computer Graphics Forum **26**(2) (2007) 143–156
13. Starck, J., Murtagh, F., Pirenne, B., Albrecht, M.: Astronomical image compression based on noise suppression. Publications of the Astronomical Society of the Pacific **108** (1998) 446–455
14. Li, M., Vitanyi, P.M.: An Introduction to Kolmogorov Complexity and Its Applications. Springer-Verlag, Berlin (1993)
15. Bennett, C., Gacs, P., Li, M., Vitanyi, P., Zurek, W.: Information distance. IEEE Transactions on Information Theory **44** (1998)
16. Li, M., Chen, X., Li, X., Ma, B., Vitanyi, P.: The similarity metric. IEEE Transactions on Information Theory **50**(12) (2004) 3250–3264
17. Cilibrasi, R., Vitanyi, P.: Clustering by compression. IEEE Trans. Information Theory **51**(4) (2005) 1523–1545
18. Cebrián, M., Alfonseca, M., Ortega, A.: The normalized compression distance is resistant to noise. IEEE Transactions on Information Theory **53**(5) (2007) 1895–1900